

# Discussion Section 7 assignment:

## Plotting graphs/charts with pylab

- As part of the next homework, you need to write code to create graphs/charts related to program running times
- This document – tiny intro to Pylab
  - Note: Pylab is part of a bigger package called matplotlib. When you look up how to do things using pylab, you'll usually find things talking about matplotlib. That's fine. The distinction is a bit mysterious to many people who manage to use them successfully.
  - **Many** Pylab/matplotlib examples here:  
<http://matplotlib.org/gallery.html>

# Pylab

- To test if pylab available, execute

```
>>> import pylab
```

If no error, you are all set

If you get an error, you need a Python installation like Anaconda (or you need to install pylab yourself – this is no fun for most people)

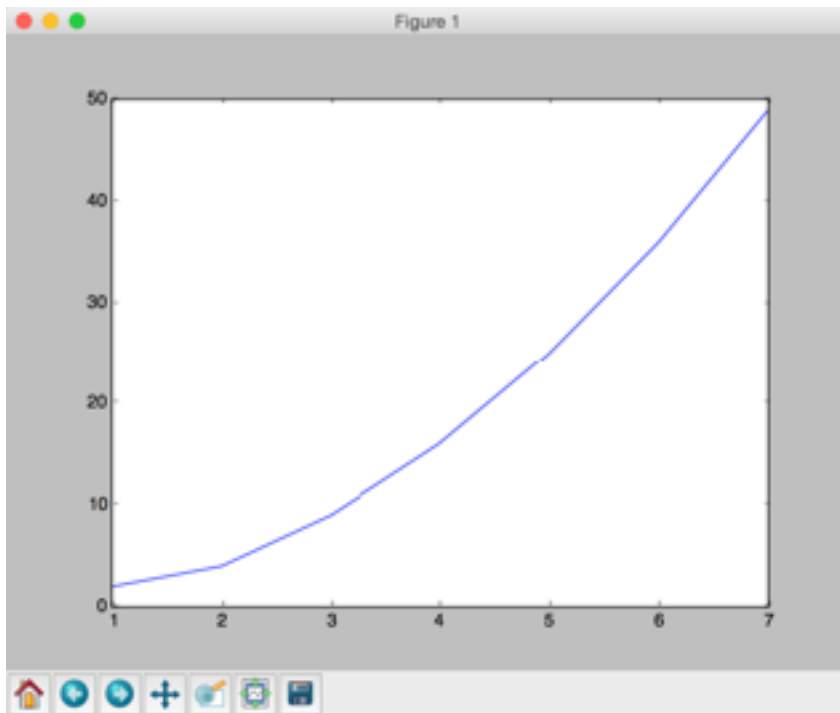
# Making simple charts with Pylab is easy

The very basics!

```
>>> pylab.plot([1, 2, 3, 4, 5, 6, 7], [2, 4, 9, 16, 25, 36, 49])
```

```
>>> pylab.show()
```

to make figure appear

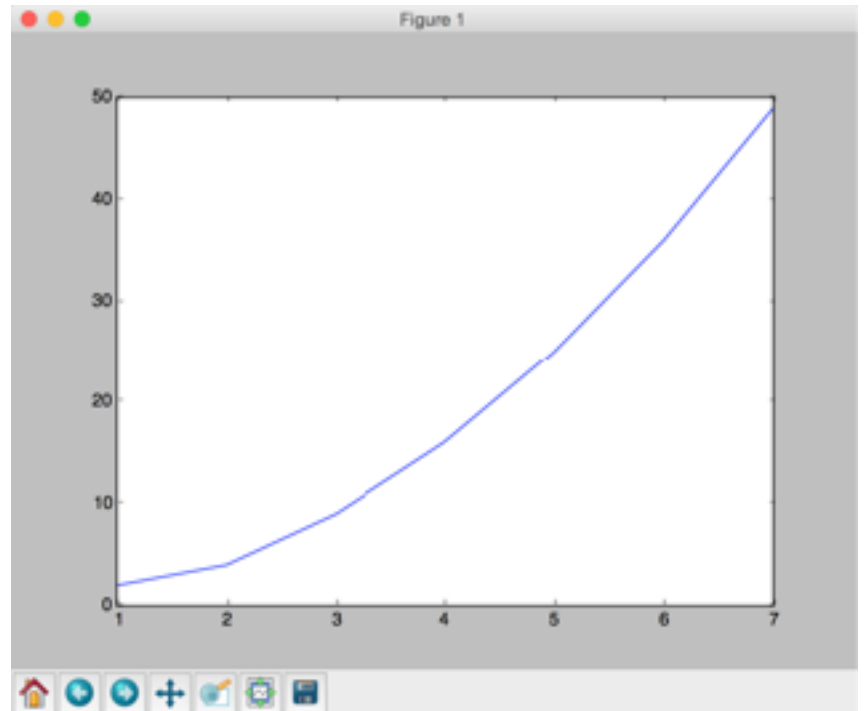


*Note: after `pylab.show()`, on some systems, you must close the window (click the little red button in upper left) to get back to the `>>>` prompt*

But better to fill lists via code. E.g. (from ds6.py)

```
def plotSquares(maxNum=20):  
    xlist, ylist = [], []  
    for x in range(1,maxNum+1):  
        xlist.append(x)  
        ylist.append(x*x)  
    pylab.plot(xlist, ylist))  
    pylab.show()
```

```
>>> plotSquares(7)
```



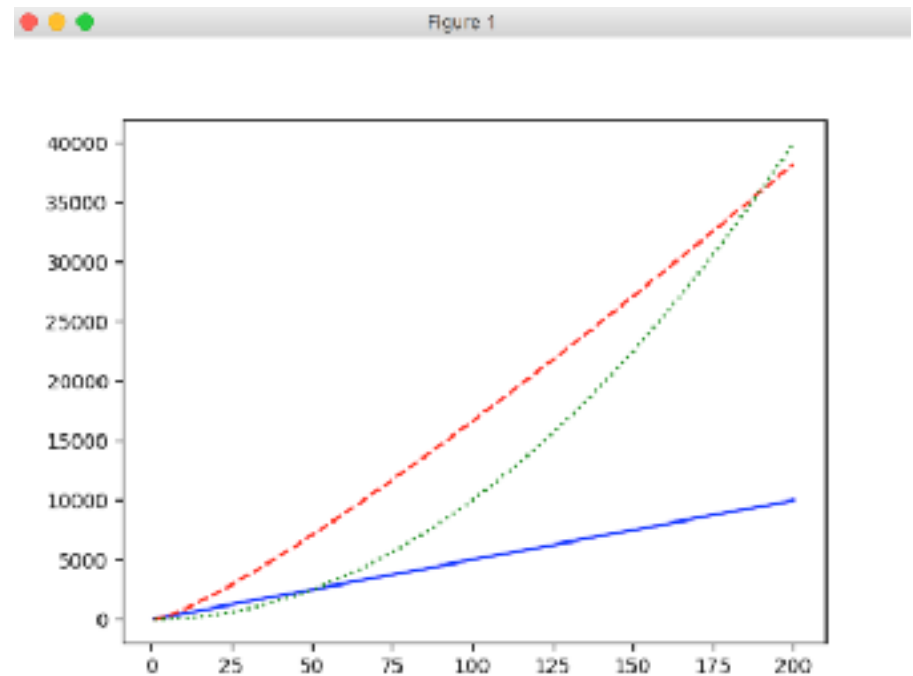
Useful commands to learn – find details online. There is a LOT of documentation online.

- `pylab.title("Title of graph")`      put title on chart
- `PyLab.plot(..., ..., linestyle = '--', color = 'g')`      plot  
using green dashed line
- `pylab.xlabel('size of problem')`      add x axis label
- `pylab.ylabel('time (in secs)')`      add y axis label
- `pylab.figure(2)`      PyLab can have figures at once.  
This says make fig. 2 the current figure.  
Subsequent commands will affect figure 2  
(until a new `pylab.figure(..)` is executed).
- `pylab.savefig('mysavedfigure')`      Save the current  
figure as a .png imagefile

```
# plot linear, n log n, and  
#quadratic functions on the  
# same chart
```

```
def plotThree(maxNum=200):  
    xlist, linlist, nlognlist, sqlist = [], [], [], []  
    for x in range(1,maxNum+1):  
        xlist.append(x)  
        linlist.append(50*x)  
        nlognlist.append(25 * x * math.log(x,2))  
        sqlist.append(x*x)  
    pylab.plot(xlist, linlist, linestyle = '-', color = 'b')  
    pylab.plot(xlist, nlognlist, linestyle = '--', color = 'r')  
    pylab.plot(xlist, sqlist, linestyle = ':', color = 'g')  
    pylab.savefig('plotTwoImage')  
    pylab.show()
```

```
>>> plotThree()
```



ds7.py

# To submit

- Write a little bit of Python code to make an interesting graph or chart. Do something different than, not just a *tiny* modification of an example in ds7.py
  - For example, define an interesting function (not just  $n$ ,  $n*n$ , or similar). Perhaps combinations of sines, cosines, logarithms, ... whatever
  - And/or plot several things in one chart using different colors and/or line styles
  - And/or use a different chart type other than the two in ds7 – line graph and bar chart. There are many more types.
- Submit (to DS7 assignment on ICON)
  1. .py file of your code
  2. an image of a chart displayed by your code